Faculty of Engineering and Technology
Electrical and Computer Engineering Department

ENCS3320-Computer Networks

Project#1

Prepared by:

Aws Shaheen	**1212585**	**sec. 2**
Alhasan Manasrah	**1211705**	**sec. 2**
Ghassan Qandeel	**1212397**	**sec. 1**

Instructors: Mohammad Jubran & Abdalkarim Awad

Deadline: 22-12-2023

Table of Contents

Part 1:

1) In your own words, what are ping, tracert, nslookup, and telnet:

Ping: is a computer network administration software utility used to test the reachability of a host on an Internet Protocol network.

Tracert: computer network diagnostic commands for displaying possible routes and measuring transit delays of packets across an Internet Protocol network.

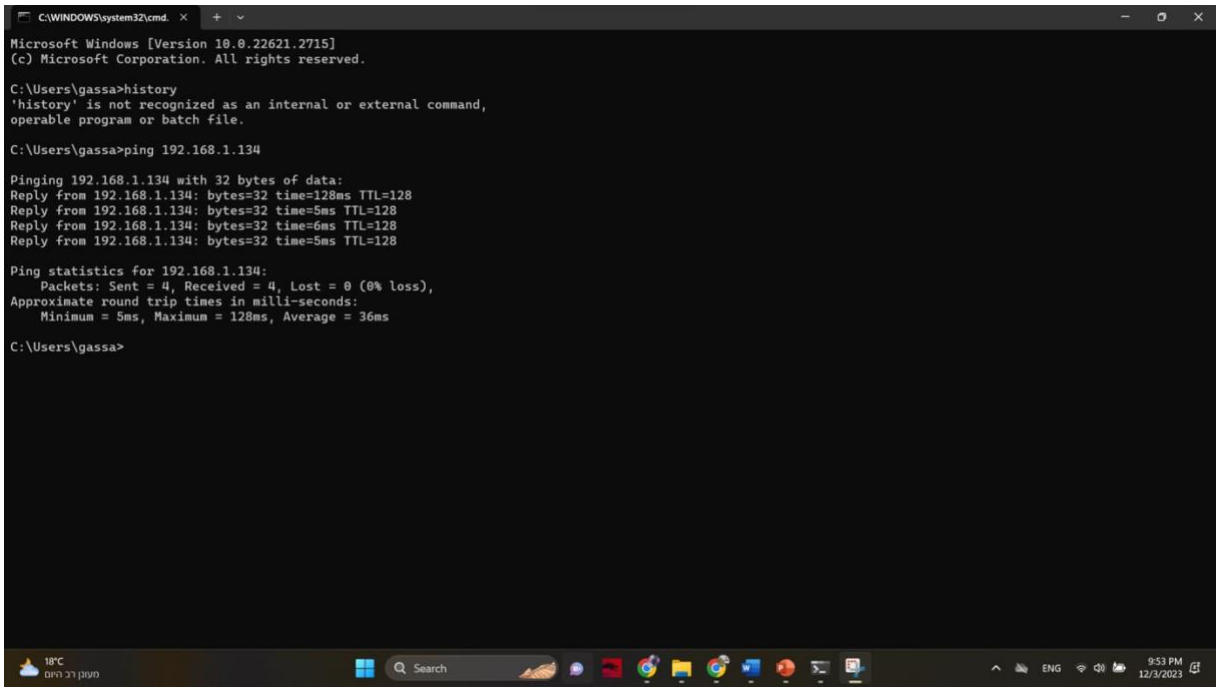
Nslookup: is a network administration command-line tool for querying the Domain Name System to obtain the mapping between domain name and IP address, or other DNS records.

telnet: is a client/server application protocol that provides access to virtual terminals of remote systems on local area networks or the Internet.

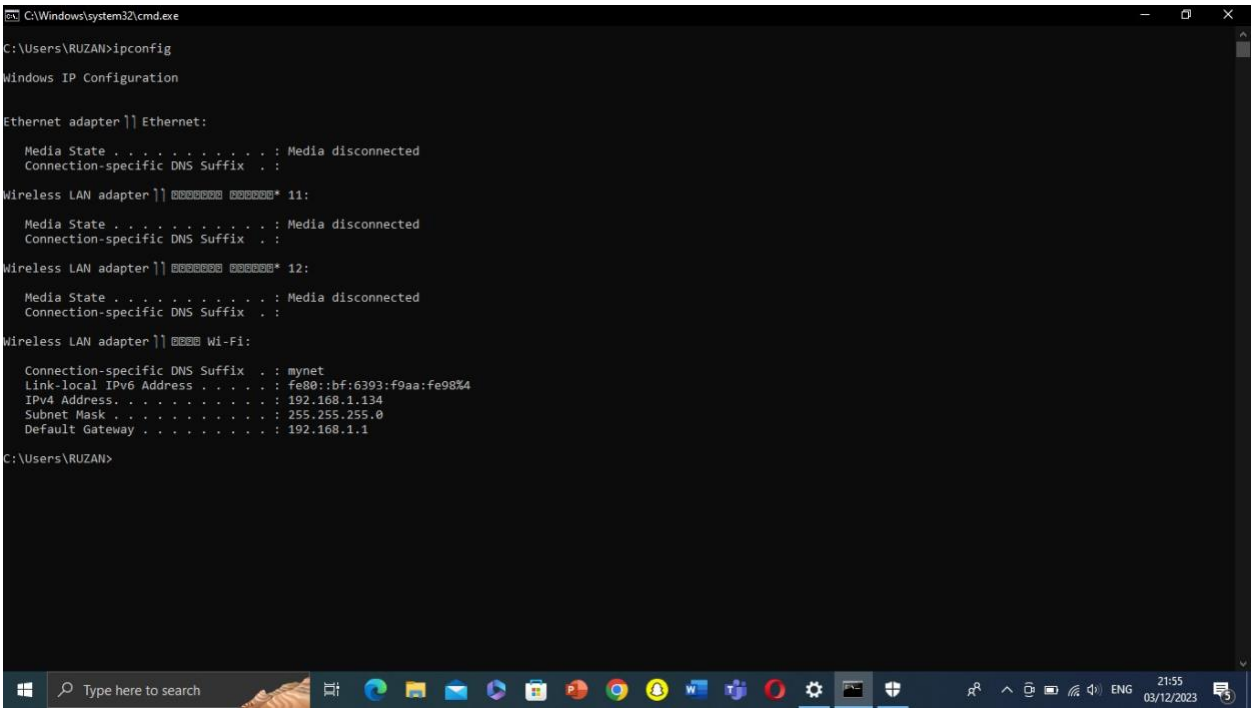
2)Make sure that your computer is connected to the internet and then run the following commands:

a)Ping a device in the same network, e.g. from a laptop to a smartphone:

Here is the result when we requested from first device to second device by writing ping 192.2.134 on the cmd line window.

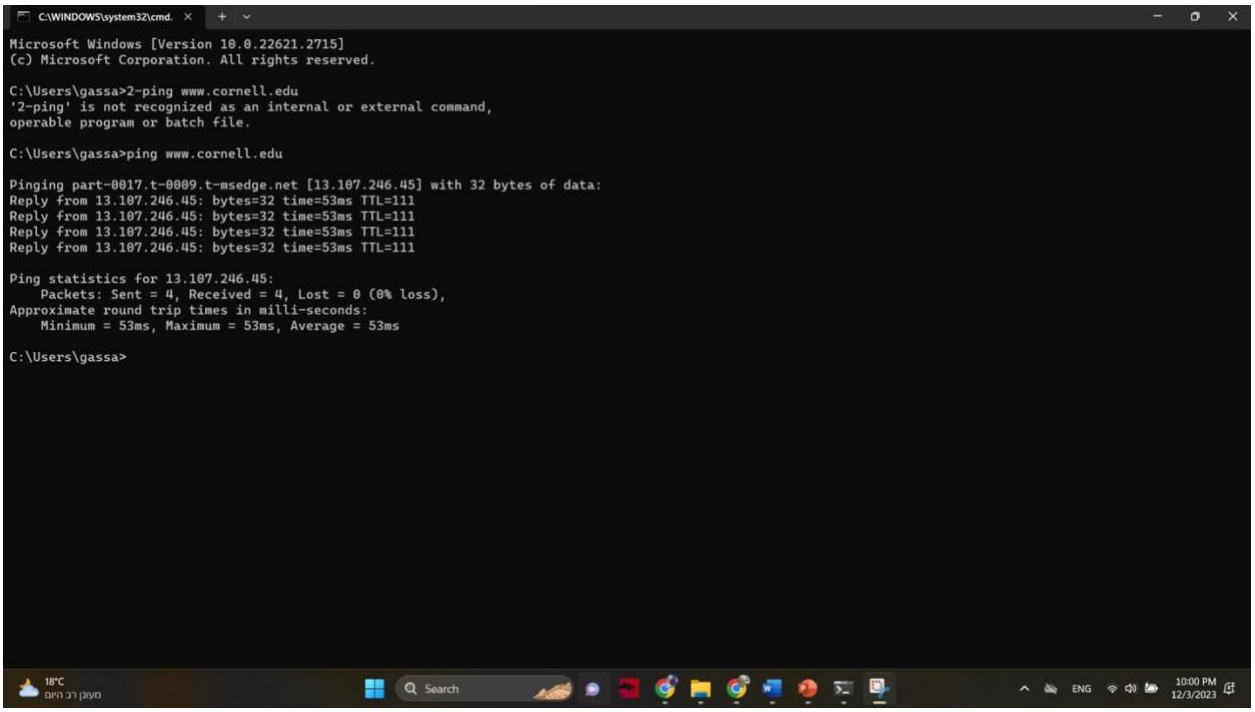


Here is the result responding from the second device by writing ipconfig command on cmd line window.



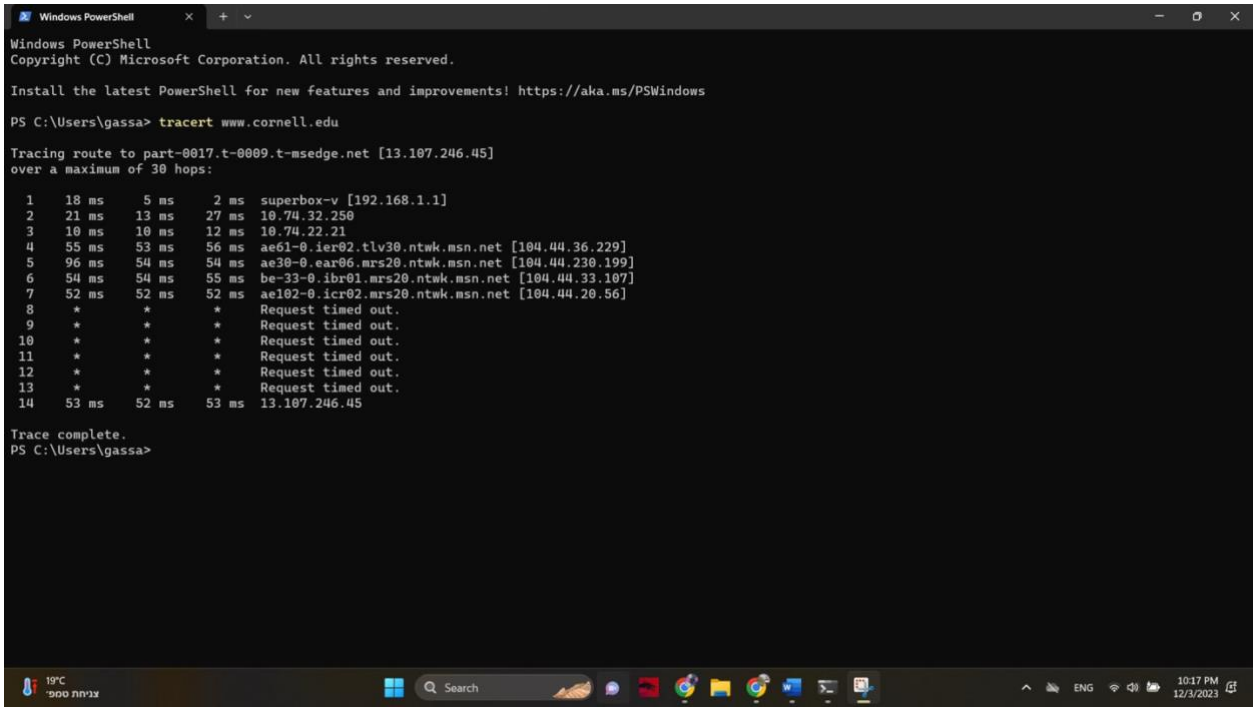
b) ping www.cornell.edu:

When you execute the "ping www.cornell.edu" command in the command line, it initiates the ICMP (Internet Control Message Protocol) Echo Request process by sending a sequence of packets to the domain "www.cornell.edu." Initially, the domain is translated into an IP address through DNS resolution. Upon reaching the target server, the packets are returned to your machine, and the time taken for the round-trip is recorded. The output provides information on the number of packets sent, received, and lost, along with statistics on the minimum, maximum, and average round-trip times measured in milliseconds. This utility is valuable for assessing network connectivity and latency between your computer and the Cornell University website.



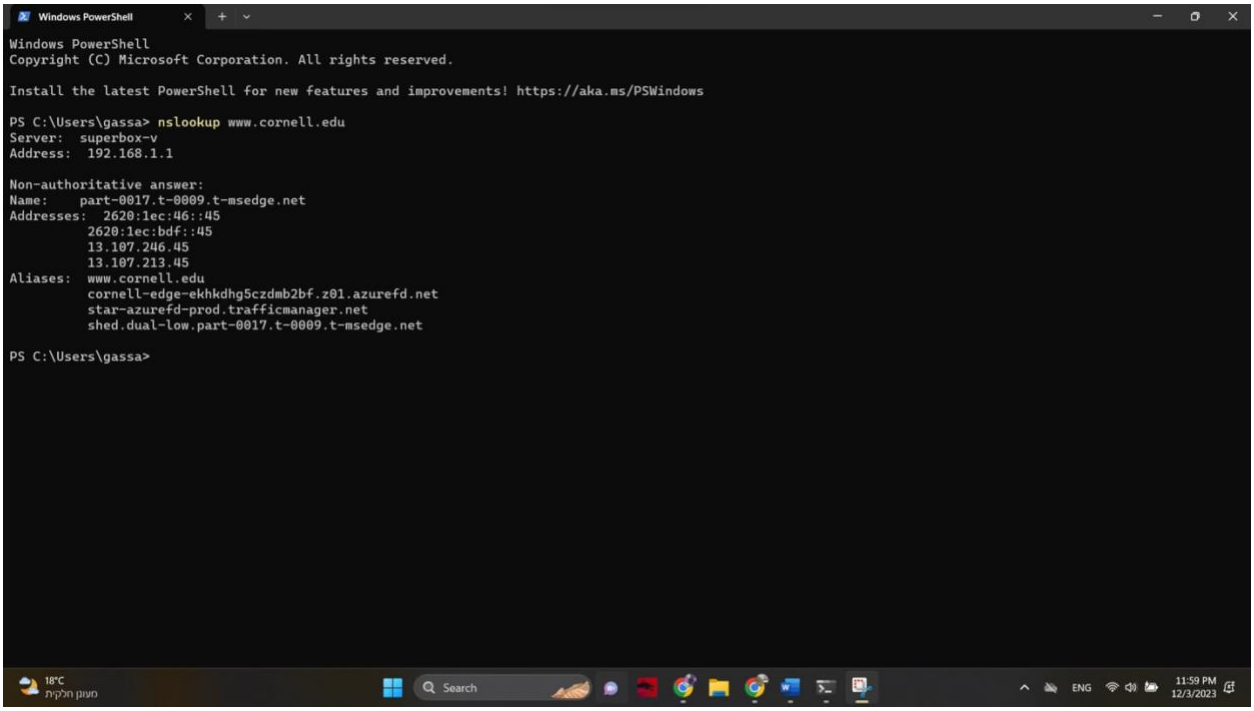
c) tracert www.cornell.edu:

Executing the "tracert www.cornell.edu" command in the command line reveals the network path taken by data packets from your computer to the Cornell University website server. This utility systematically sends packets to each hop, typically a router or switch, along the route, documenting the round-trip time for each leg of the journey. The output provides a comprehensive list of intermediate stops (hops) the packets traverse, offering valuable insights for diagnosing potential network issues or identifying bottlenecks.



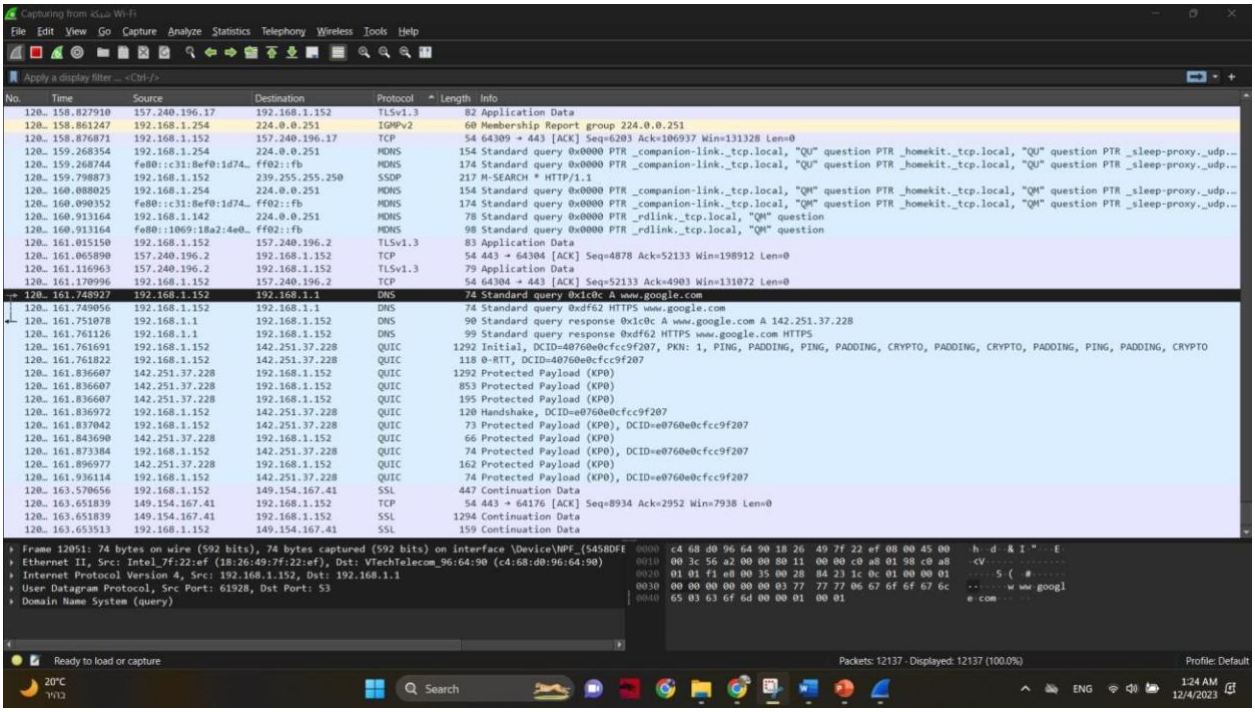
d) nslookup www.cornell.edu:

Executing the "nslookup www.cornell.edu" command in the command line involves querying a Domain Name System (DNS) server to translate the domain name www.cornell.edu into its corresponding IP address. This utility furnishes details about the DNS records linked with the domain, particularly the IP address to which the domain name is mapped. The tool is instrumental for troubleshooting DNS-related problems, confirming domain configurations, or obtaining insights into domain addresses.



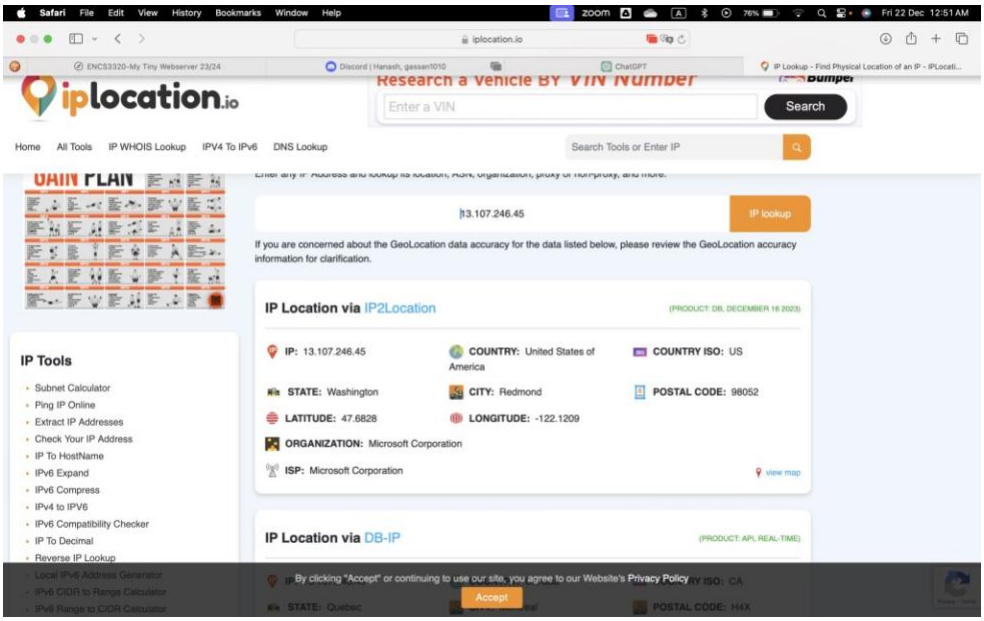
3)use wireshark to capture some DNS messages:

Once you've captured DNS messages, you can analyze the packets to see DNS query and response details. Look for DNS packets and inspect the query and response sections. Regarding the ping results and determining if the response is from the USA, ping response times alone are not reliable indicators of geographical location. While longer response times might suggest greater distance.



- From the ping results, do you think the response you have got is from USA? Explain your answer briefly.

After looking up for the IP address using “https://iplocation.io/ip/13.107.246.45” website that we got after applying “ping www.cornell.edu” command in cmd line window, here is the result:

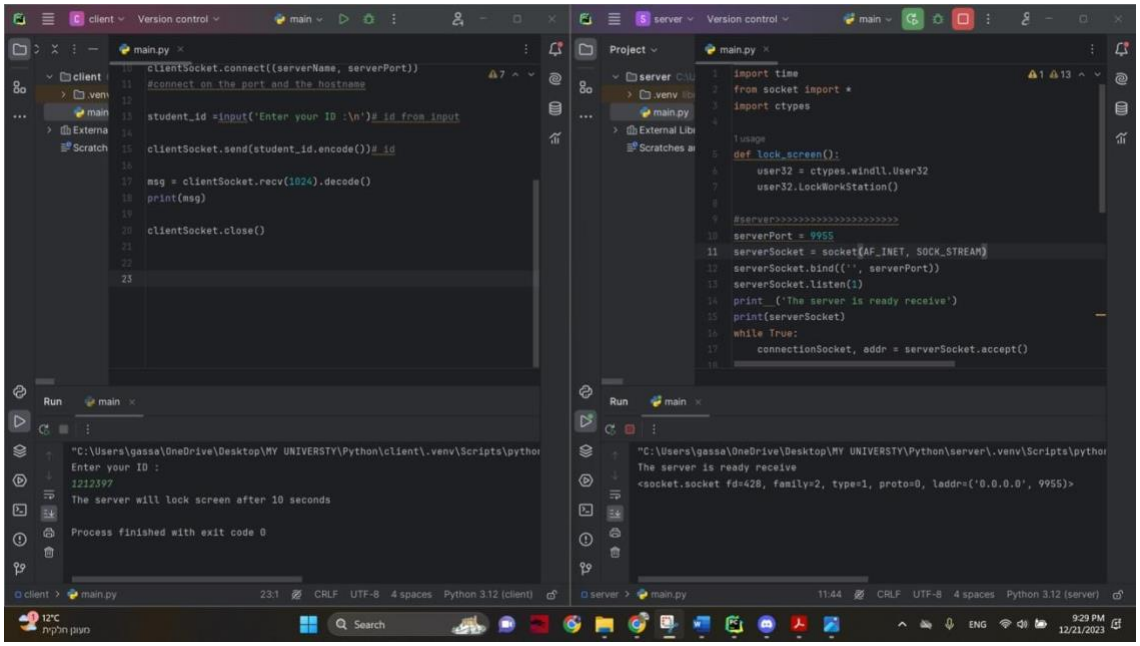


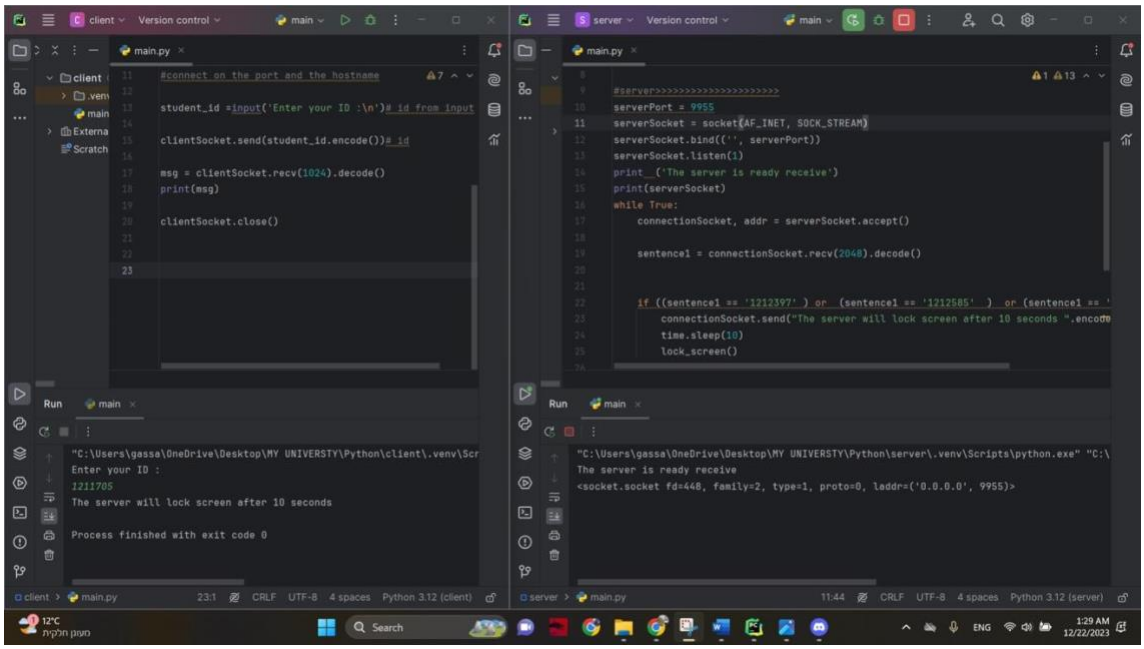
As we can see the response is from USA.

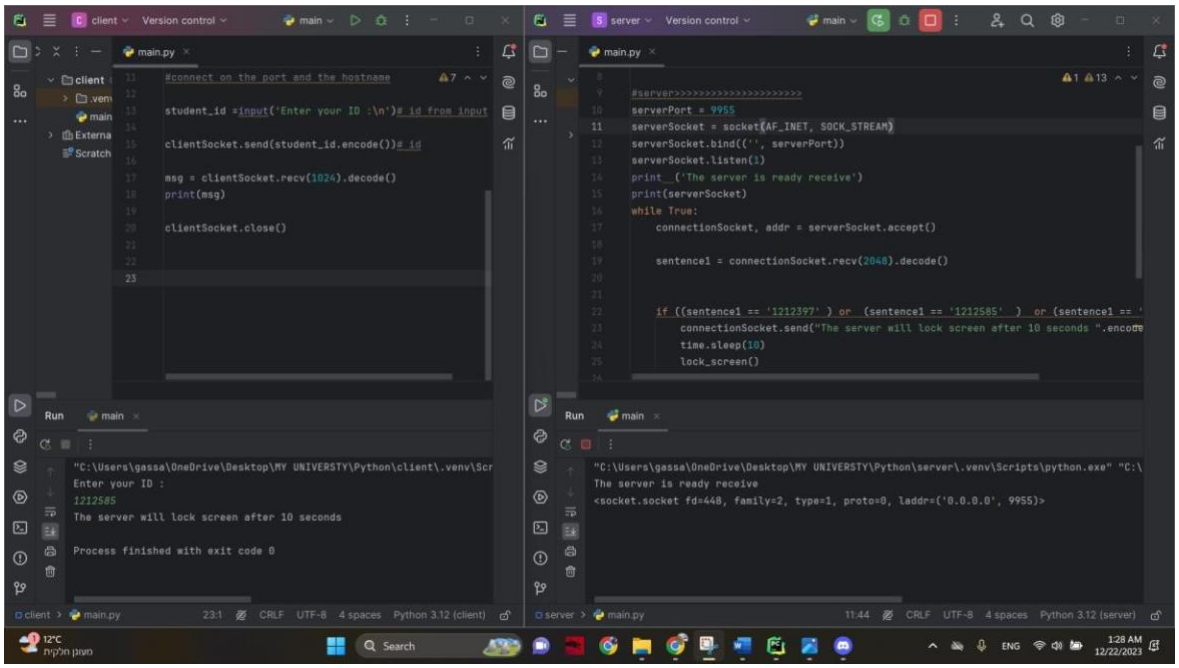
Part2: socket programing using python:

In this part we have used socket programing to implement TCP client and server applications using python, so in this program the client sends a message to the server if the message contains any of our ID's it will display a message that the screen will lock after 10 seconds and lock it after 10 seconds, if it's not it will display an error message on the server side without locking the screen.

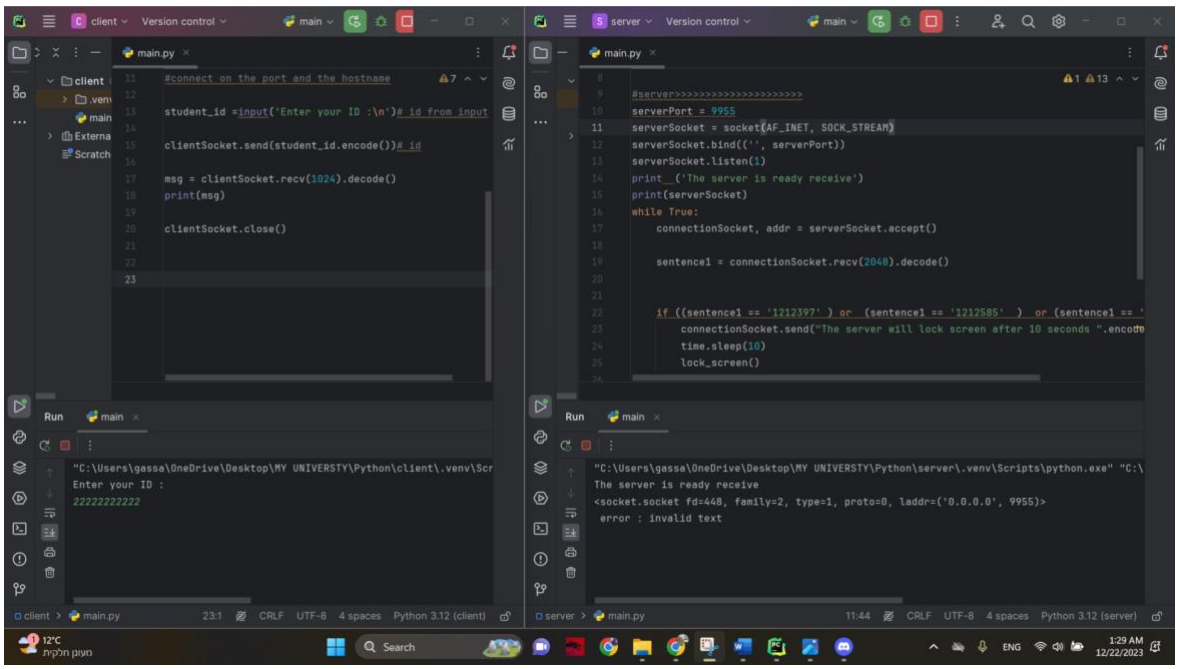
Those are 3 successful runs with our 3 different ID's:







And this is a run gives an error because we didn't entered on of our ID's:



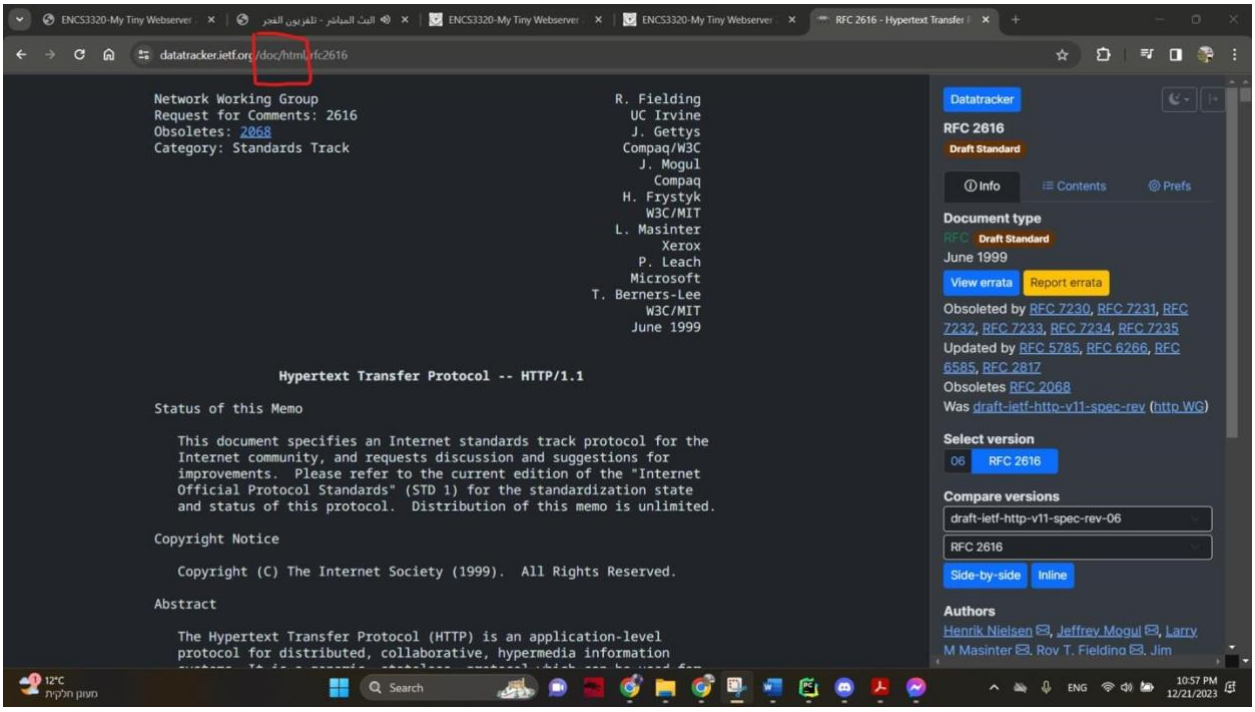

```
#client from socket
import * serverName =
'localhost' serverPort =
```

```
9955 #define name and the
port

#client >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

clientSocket = socket(AF_INET, SOCK_STREAM)
# create the TCP client server
clientSocket.connect((serverName, serverPort))
#connect on the port and the hostname

student_id =input('Enter your ID :\n')# id from input

clientSocket.send(student_id.encode())# id

msg = clientSocket.recv(1024).decode()
print(msg)
clientSocket.close()

 #server
import time from
socket import *
import ctypes
 def
lock_screen():
    user32 = ctypes.windll.User32
user32.LockWorkStation()
#server>>>>>>>>>>>>>>>>>>>>>> serverPort
= 9955
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('', serverPort)) serverSocket.listen(1)
print  ('The server is ready receive')
print(serverSocket) while True:
    connectionSocket, addr = serverSocket.accept()

    sentence1 = connectionSocket.recv(2048).decode()
      if ((sentence1 == '1212397' ) or  (sentence1 == '1212585'  )
or
(sentence1 == '1211705'  )):         connectionSocket.send("The
server will lock screen after 10 seconds ".encode())
time.sleep(10)         lock_screen()

else:
        print(" error : invalid text")
connectionSocket.close()
```

part3:

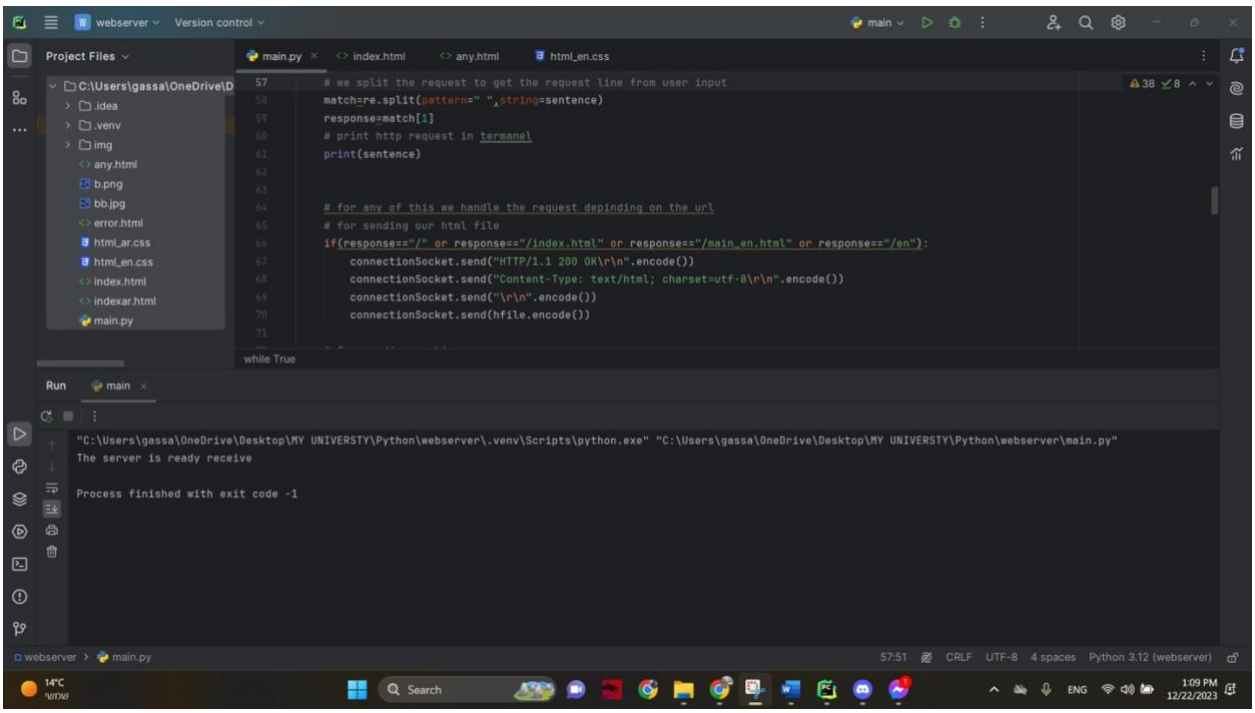
0) *from rfce2616, what is Content-Type in the HTTP request and why do we need it?*

The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient or, in the case of the HEAD method, the media type that would have been sent had the request been a GET, and we need it Because it tells the server what kind of data is being sent in the request so that the server can properly interpret and process it, it is crucial for both the client and server to understand how to handle the data being exchanged.

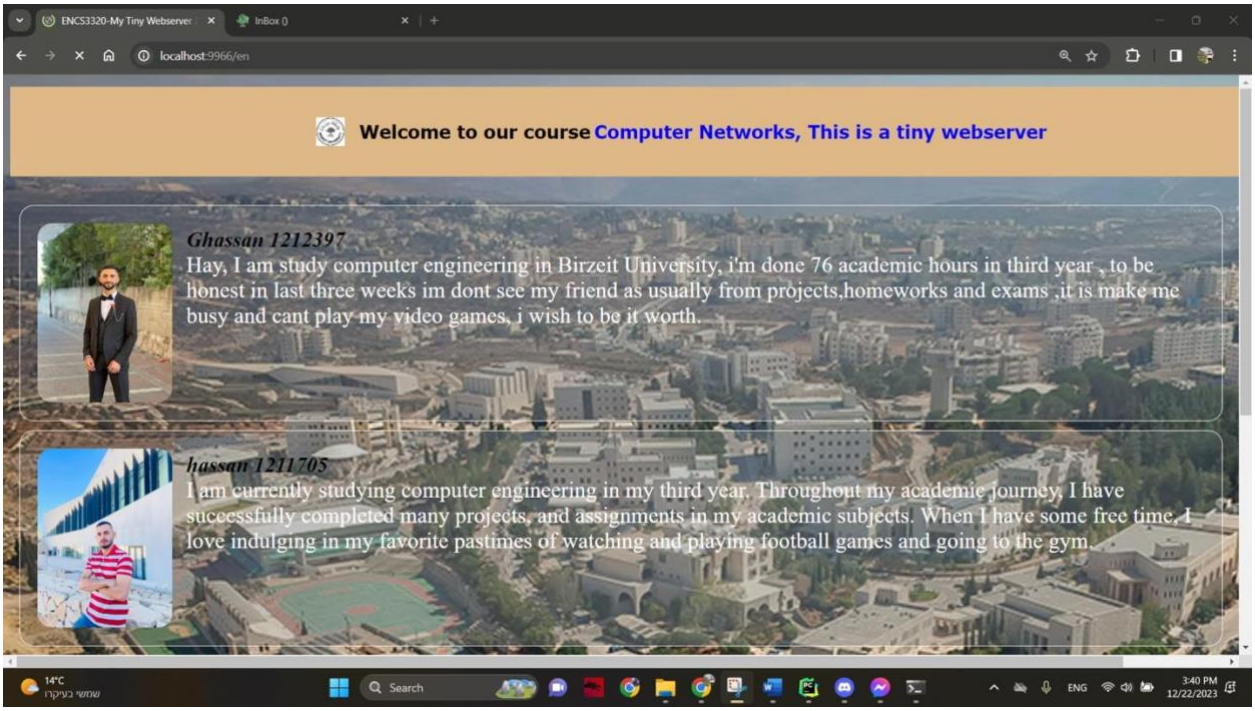


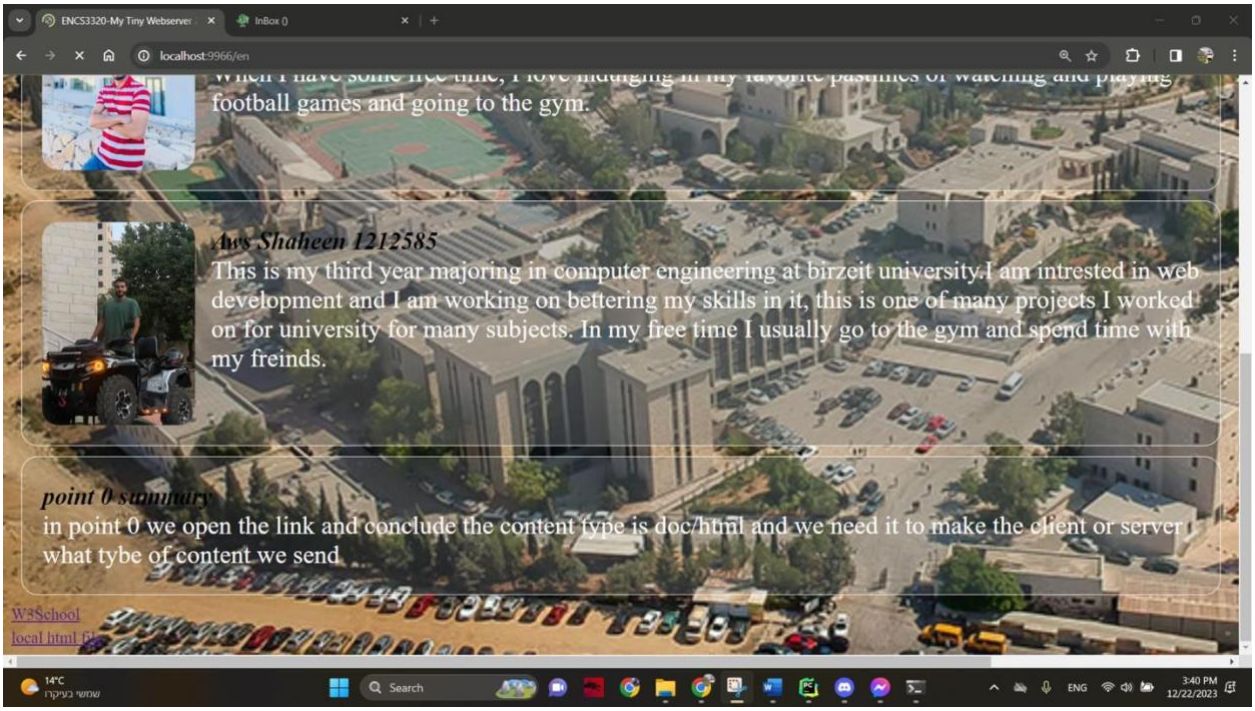
1) if the request is / or /index.html or /main_en.html or /en (for example localhost:9966/ or localhost:9966/en) then the server should send main_en.html file with Content-Type: text/html:

in this part if the request was any one of the above requests so the server send the English html web bage



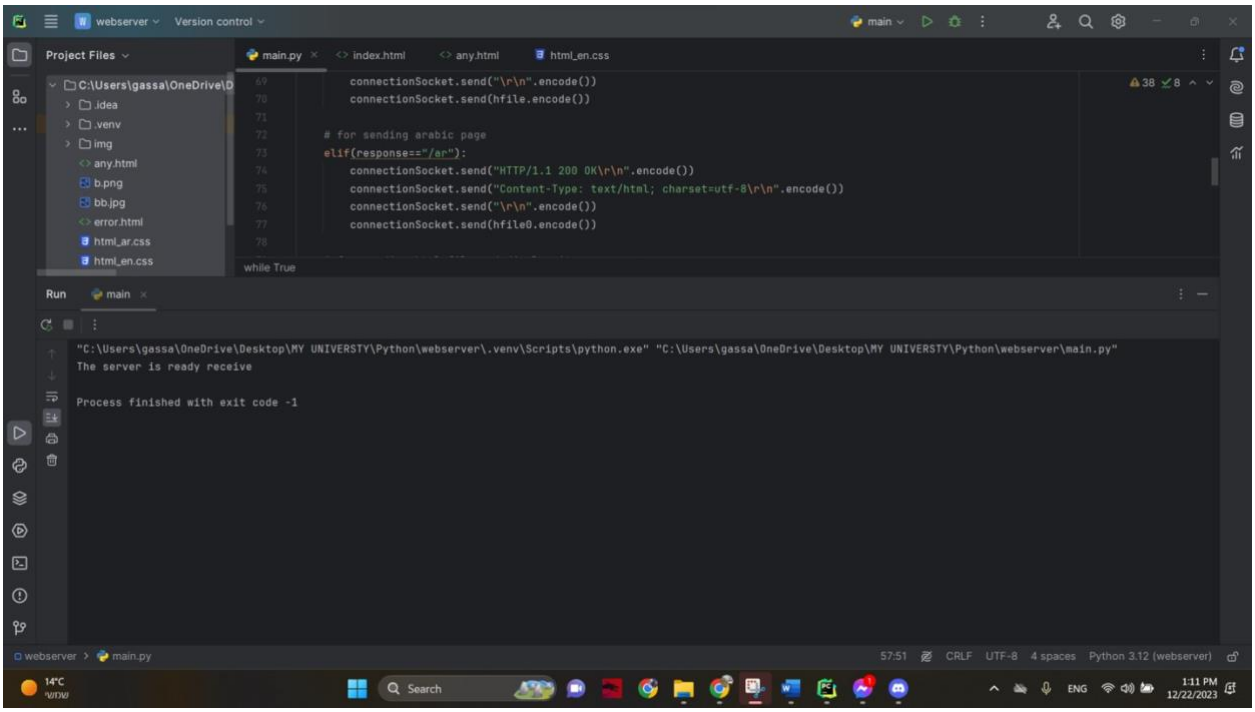
This is the response from server, it is our website bage:



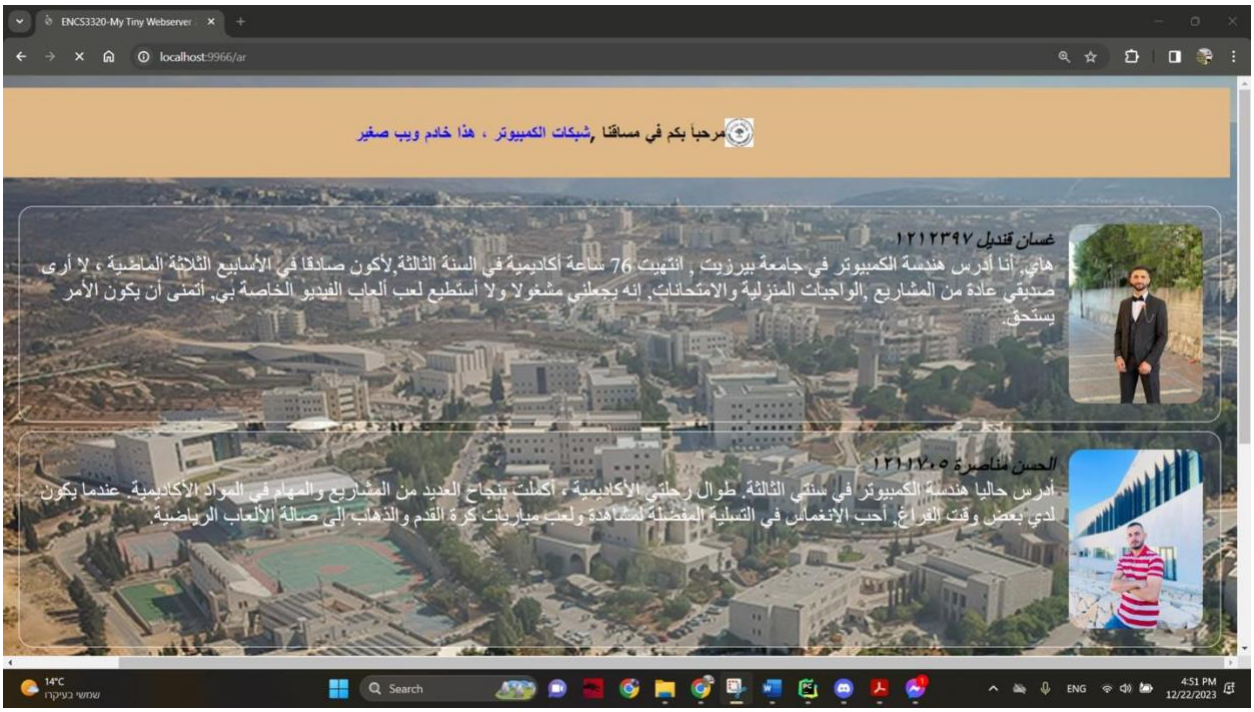


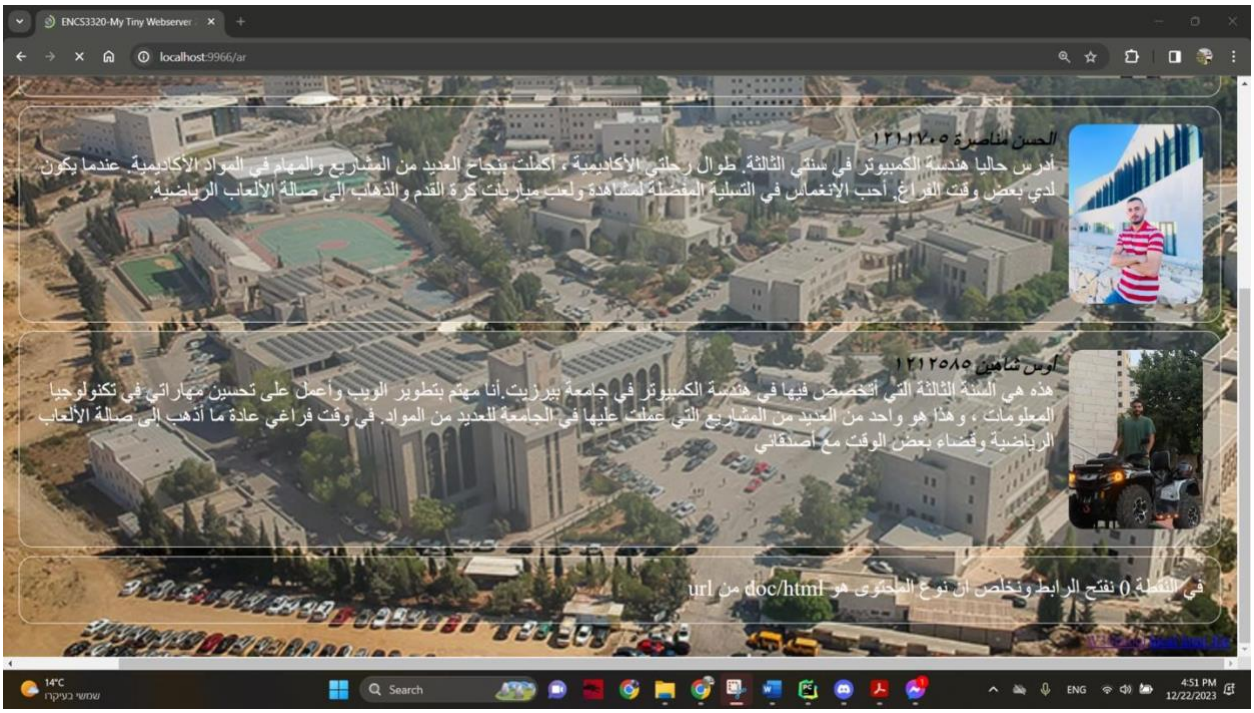
2)If the request is /ar then the server response with main_ar.html which is an Arabic version of main_en.html:

In this part if the request was /ar then the response from server should be the Arabic html web bage

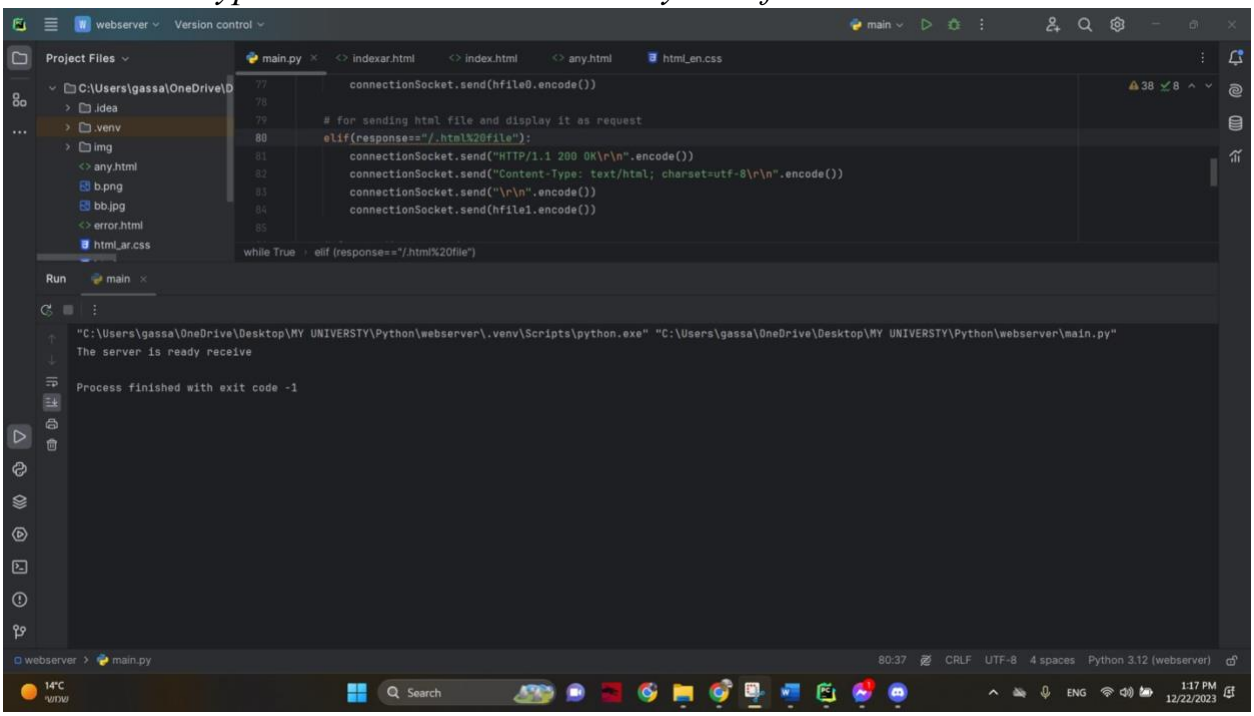


This is the response we got from server it is our website page :



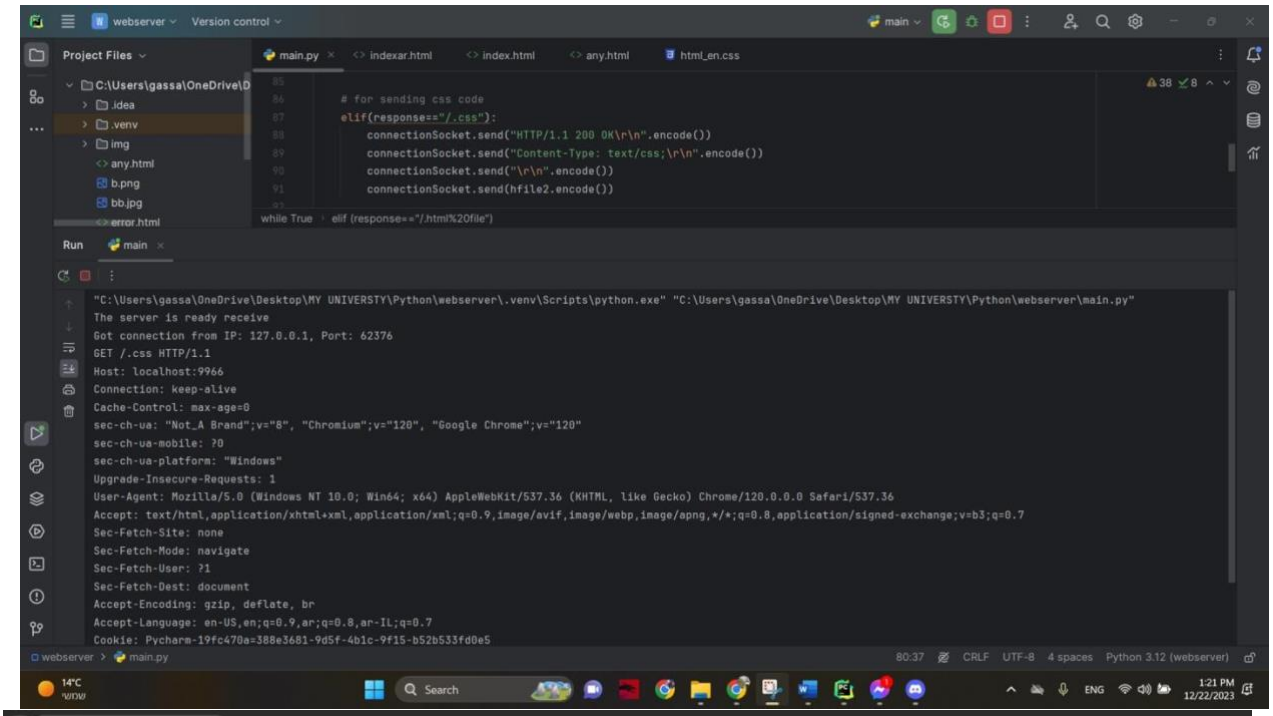


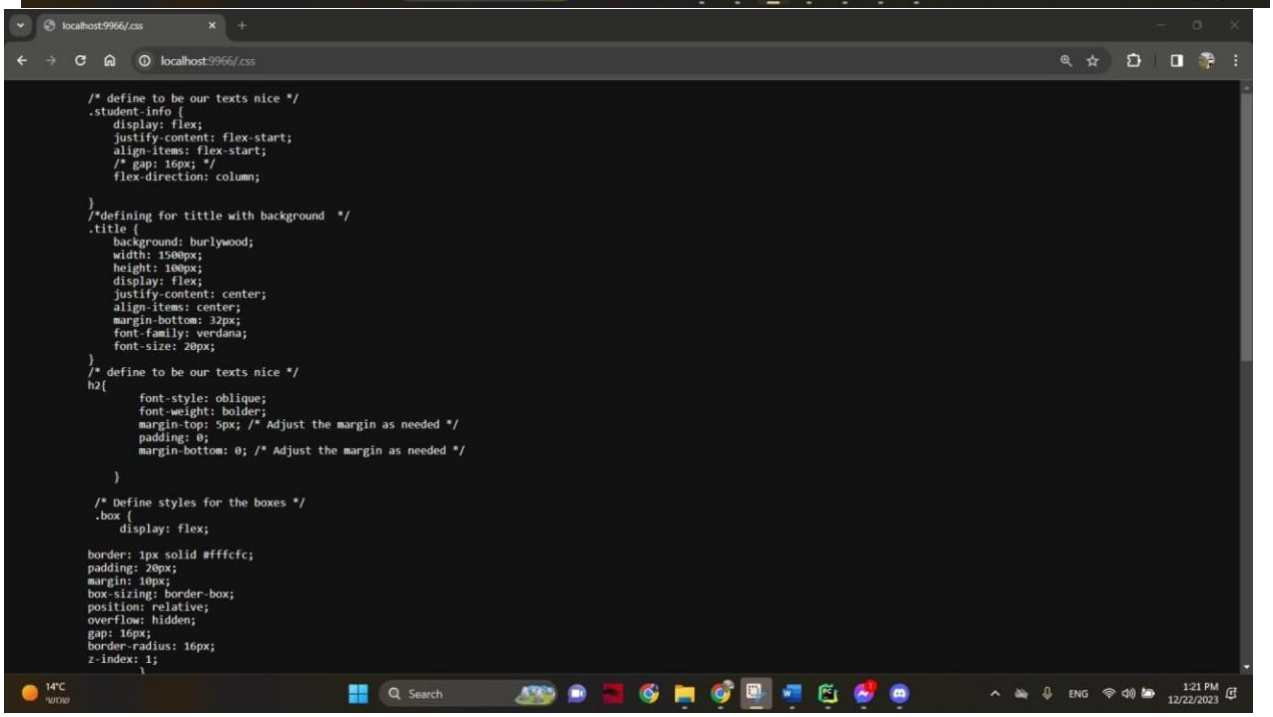
3) if the request is an .html file then the server should send the requested html file with Content-Type: text/html. You can use any html file:



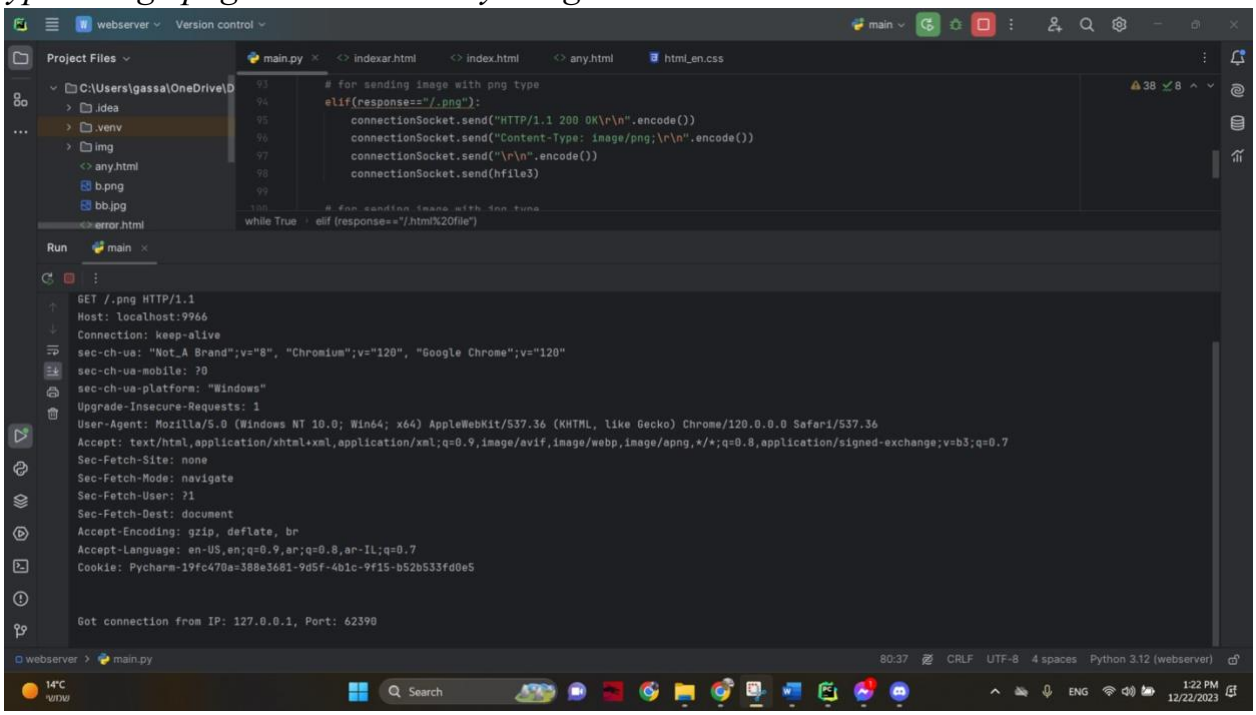
The response is this simple html server:

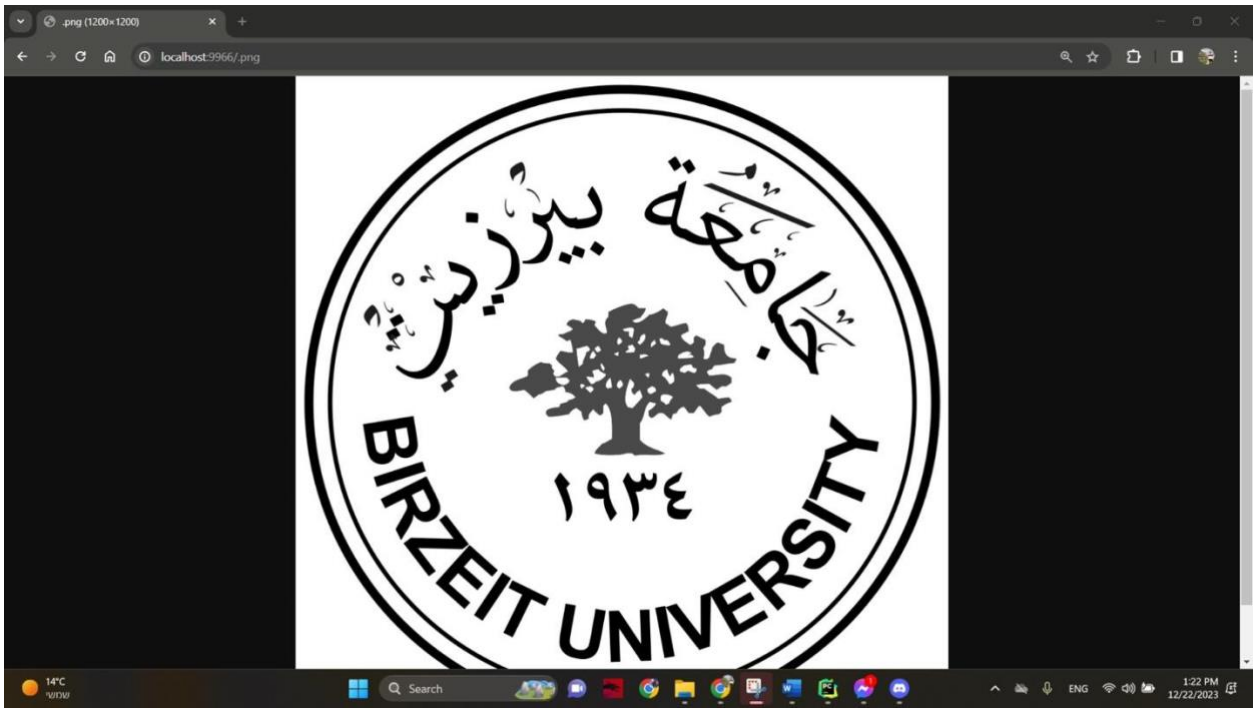
4) if the request is a ***.css*** *file then the server should send the requested css file with Content-Type: text/css. You can use any CSS file:*



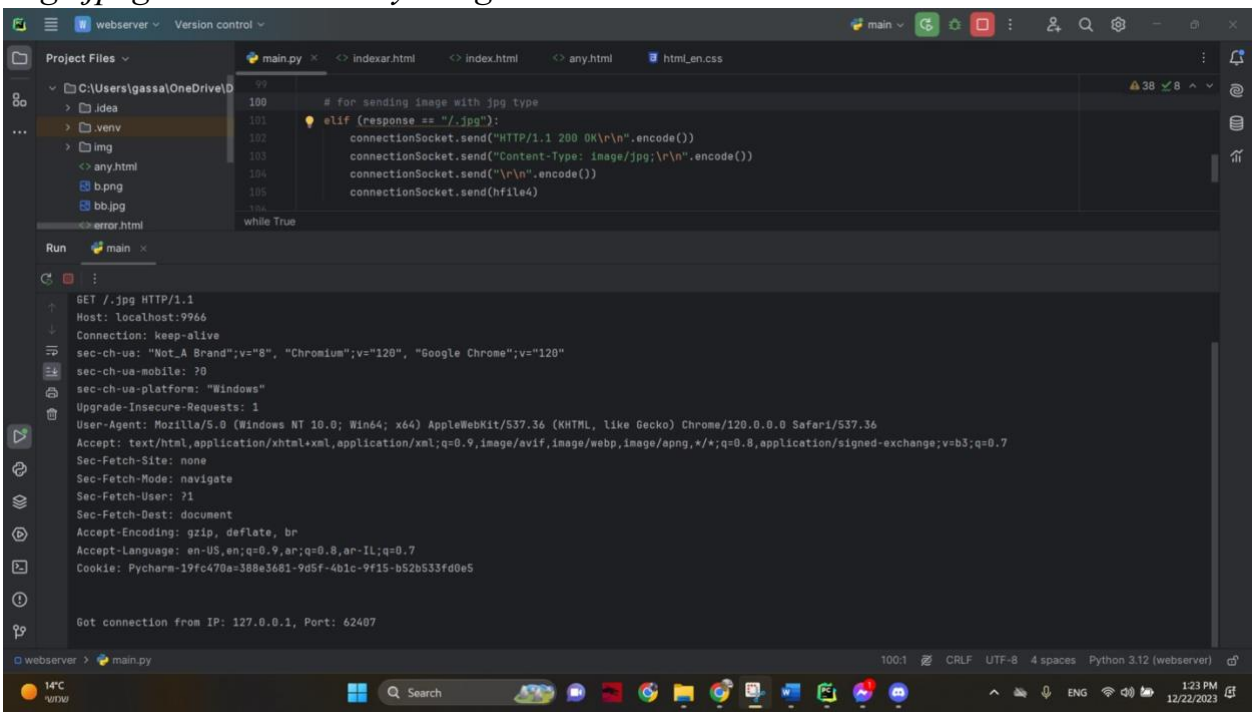


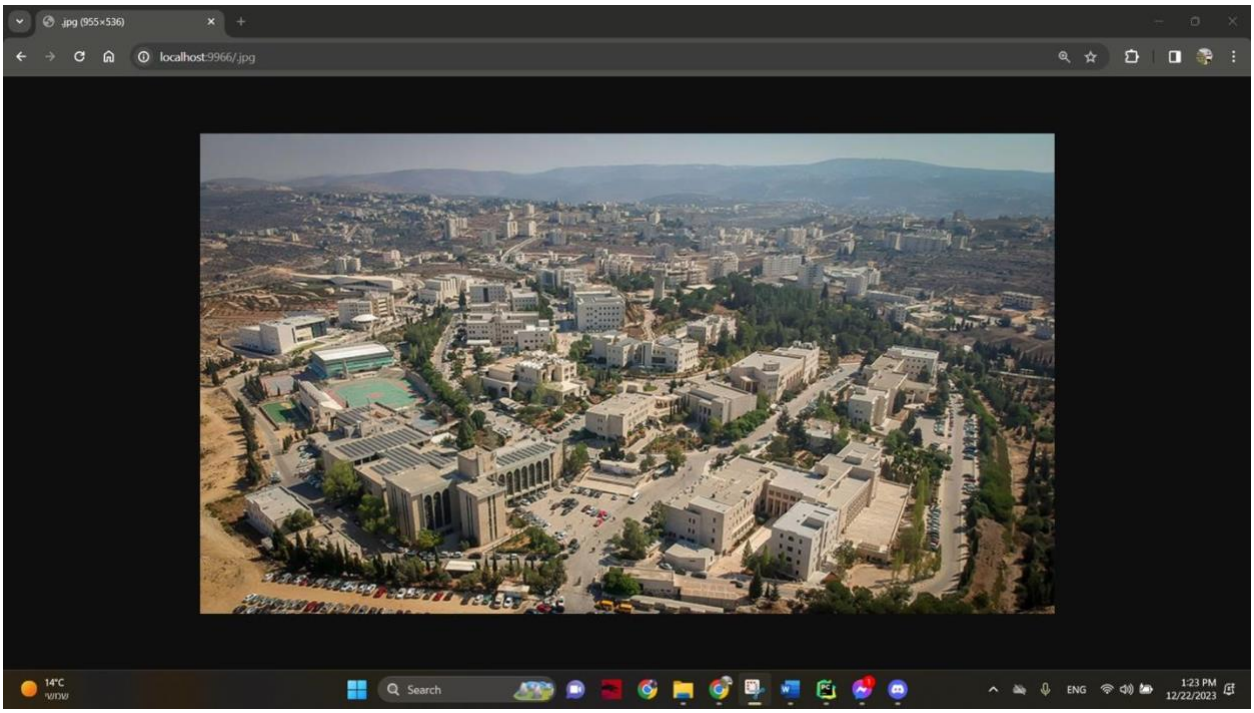
*5) if the request is a .**png** then the server should send the png image with Content-Type: image/png. You can use any image*





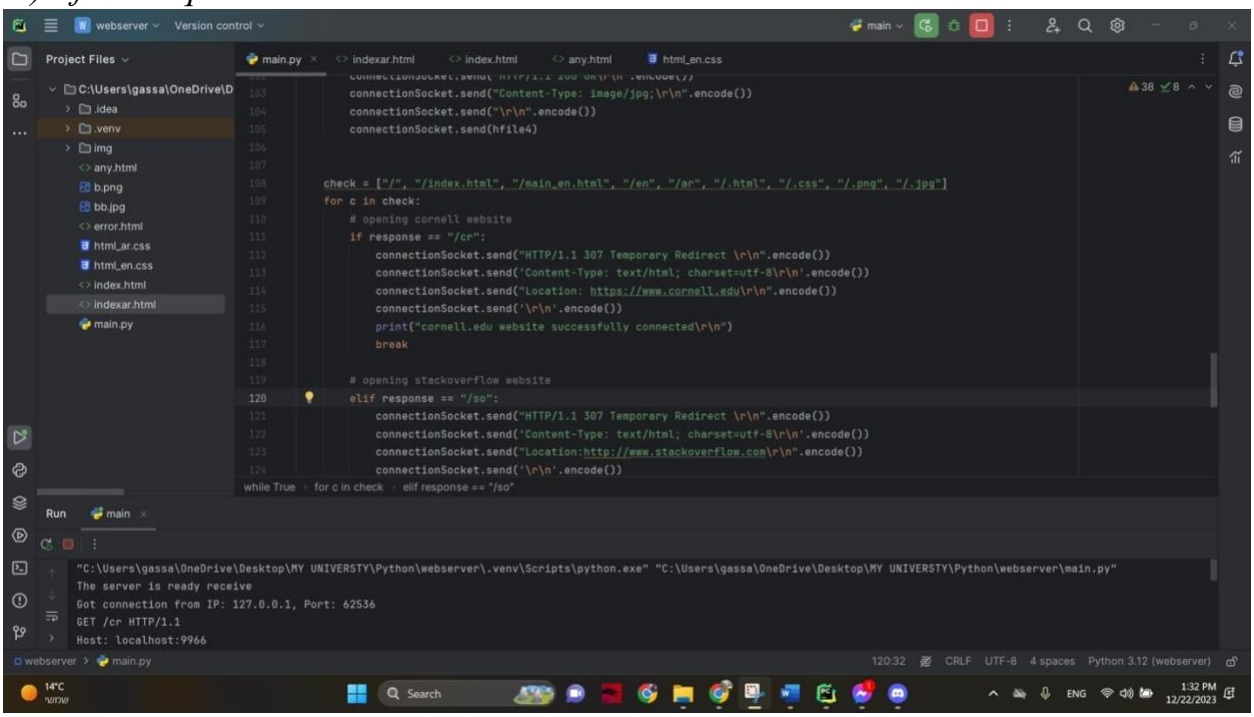
*6) if the request is a **.jpg** then the server should send the jpg image with ContentType: image/jpeg. You can use any image:*

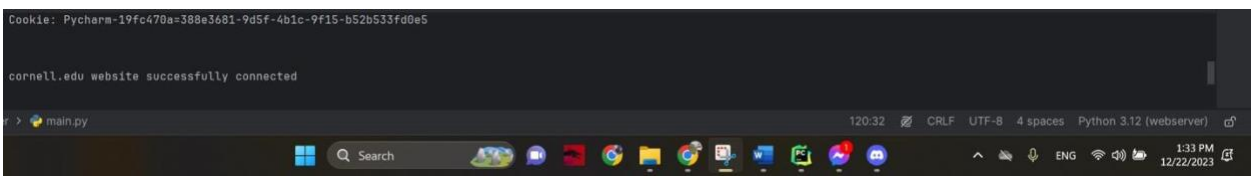


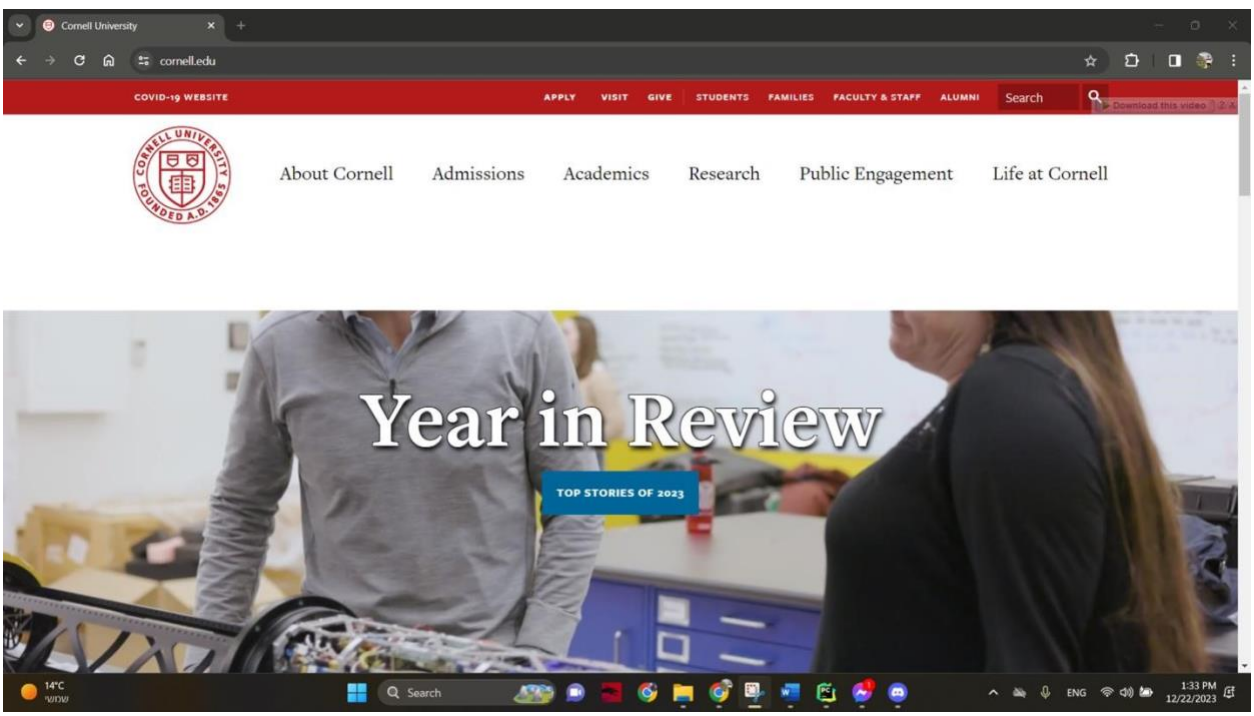


7) Use the status code 307 Temporary Redirect to redirect the following

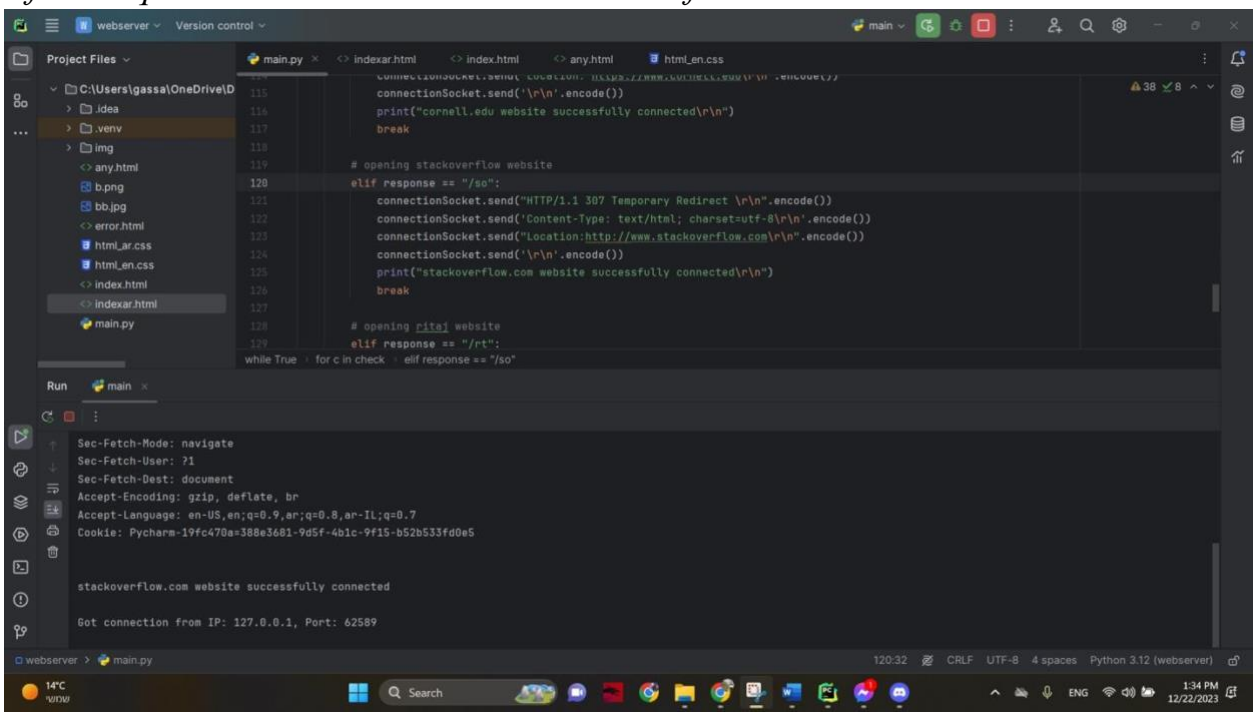
a) If the request is /cr then redirect to cornell.edu website:

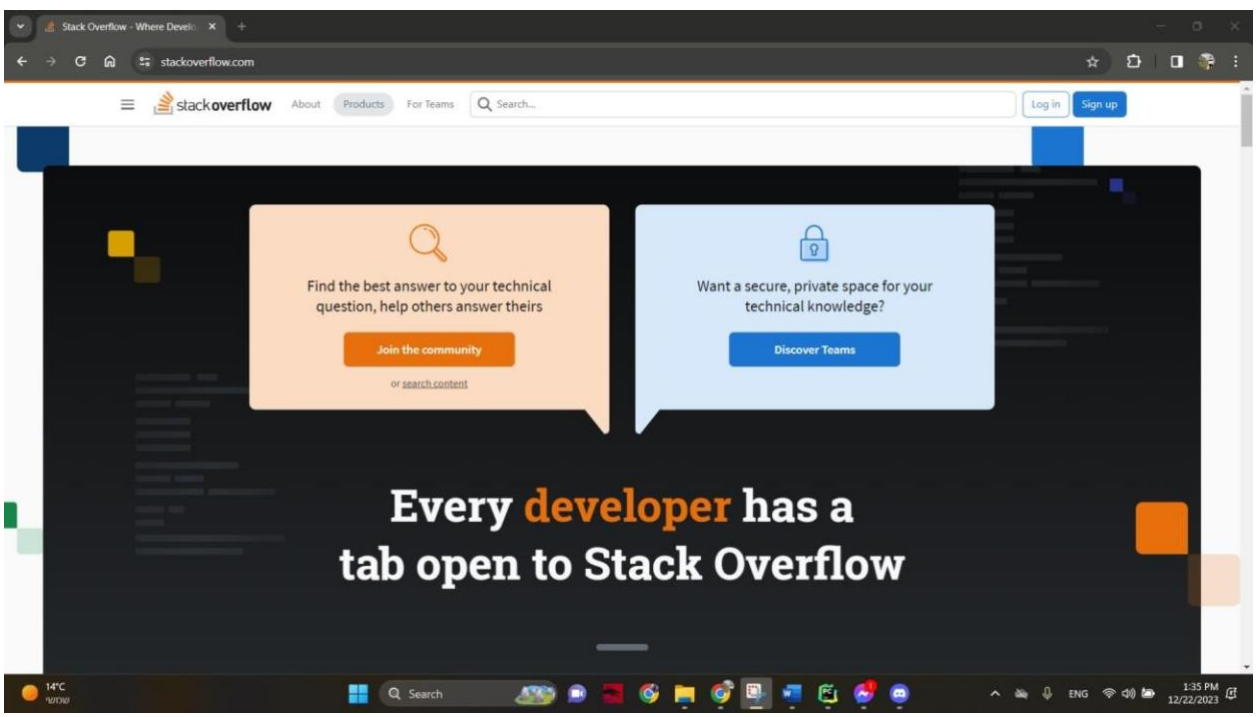




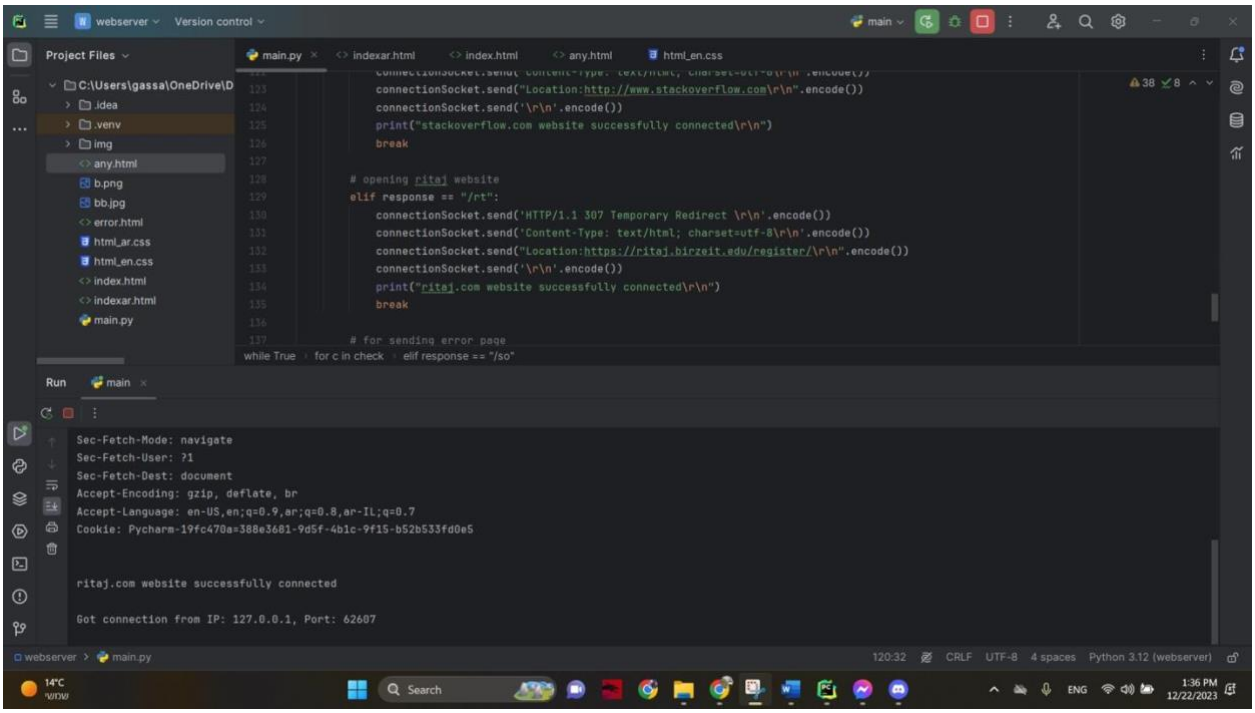


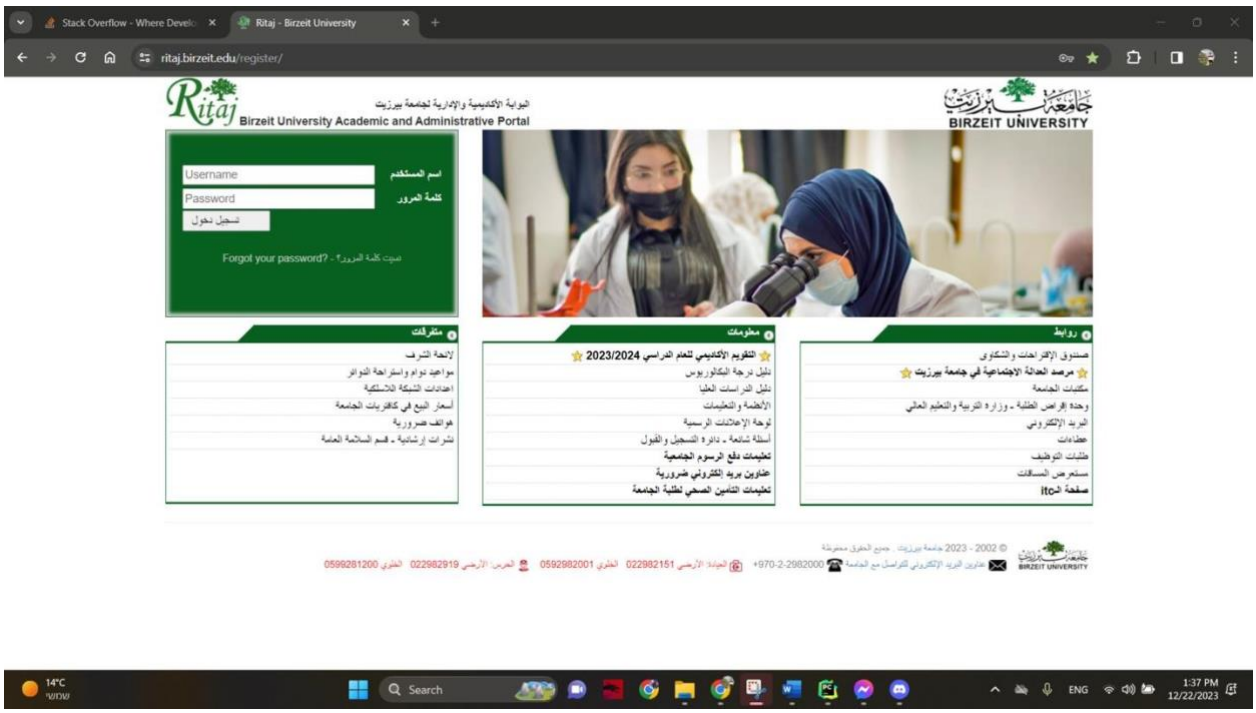
b) If the request is /so then redirect to stackoverflow.com website:



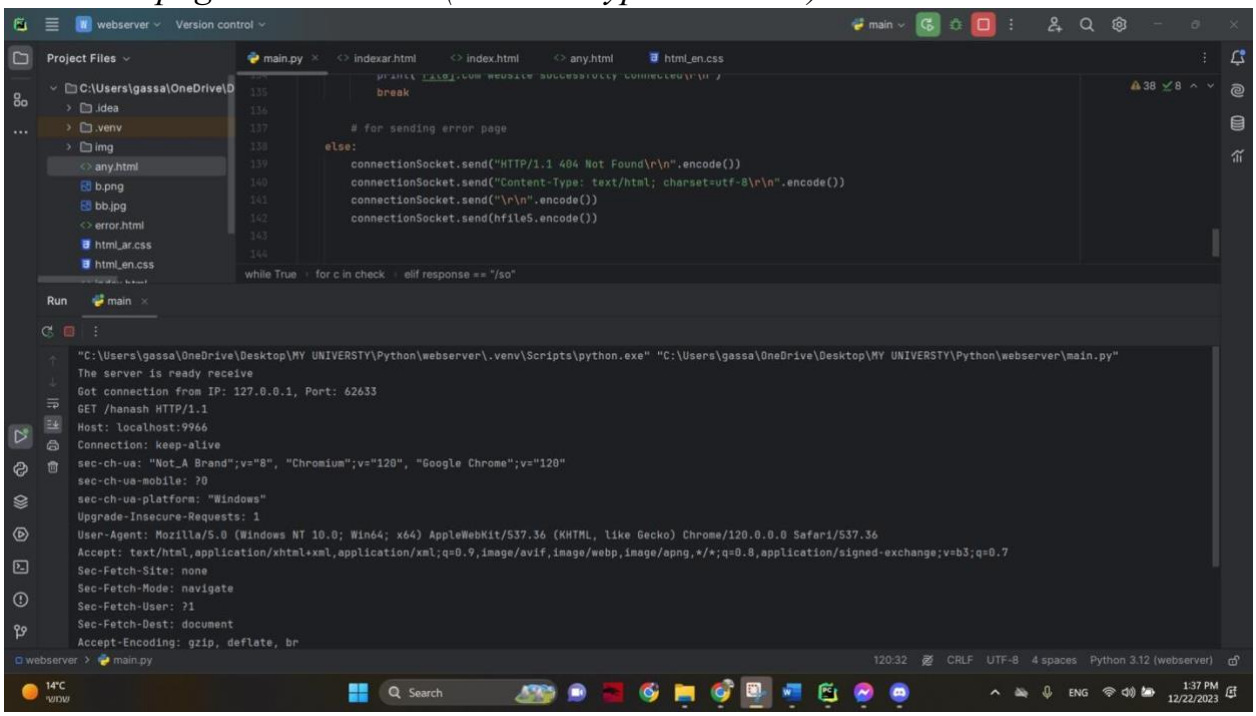


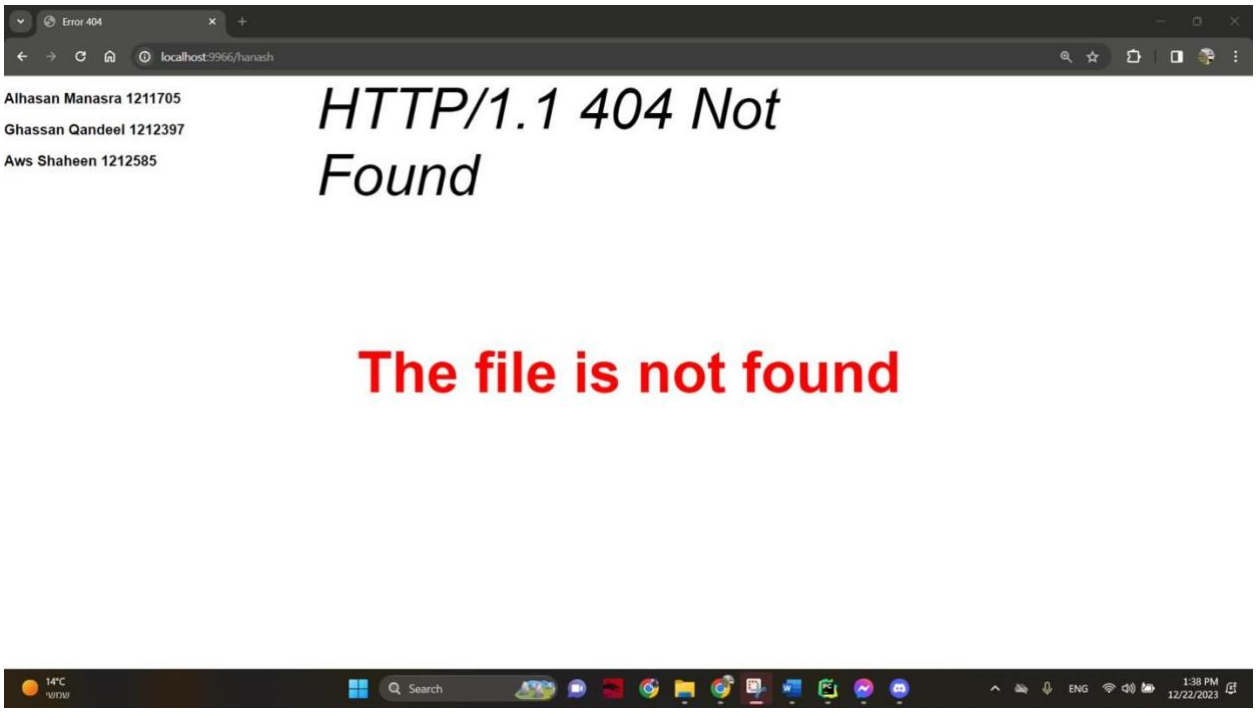
c) If the request is /rt then redirect to ritaj website:





8) If the request is wrong or the file doesn't exist the server should return a simple HTML webpage that contains (Content-Type: text/html):





Python code

```
from socket import *
import re
#server #define
server port
serverPort = 9966
#define TCP server
serverSocket = socket(AF_INET, SOCK_STREAM) #
make binding with any ip address by ''
serverSocket.bind(('', serverPort))
# listen for requests
serverSocket.listen(1) # print for
the server is ready print('The
server is ready receive')

# open index.html file with open('index.html', 'r',
encoding='utf-8') as file:
   hfile = file.read()

# open arabic index.html file with open('indexar.html',
'r', encoding='utf-8') as file0:
   hfile0 = file0.read()

# open html file for display it with open('any.html',
'r',  encoding='utf-8') as file1:
   hfile1 = file1.read()

# open css file for display it with open('html_en.css',
'r', encoding='utf-8') as file2:
   hfile2 = file2.read()

# open image with png tybe with
open('b.png', 'rb') as file3:
   hfile3=file3.read()

# open image with jpg tybe with
open('bb.jpg', 'rb') as file4:
   hfile4 = file4.read()

# open html for error page with open('error.html', 'r',
encoding='utf-8') as file5:
   hfile5 = file5.read()
 with open('img/gf.jpg', 'rb') as
file6:
   hfile6 = file6.read()
 with open('img/hf.jpg', 'rb') as
file7:
   hfile7 = file7.read()
 with open('img/af.jpg', 'rb') as
file8:
    hfile8 = file8.read()
 with open('html_ar.css', 'r', encoding='utf-8') as
file9:
   hfile9 = file9.read()
```

```
 while
True:

    # accept all requests
    connectionSocket, addr = serverSocket.accept()
     ip = addr[0]
port = addr[1]
    print('Got connection from', "IP: " + ip + ", Port: " + str(port))

    # receve http reqeust and store it in sentence
sentence = connectionSocket.recv(4096).decode()
    # we split the request to get the request line from user input
match=re.split(pattern=" ",string=sentence)     response=match[1]
print("........."+response+".......")     # print http request in
termanel     print(sentence)


    # for any of this we handle the request depinding on the url
    # for sending our html file
    if(response=="/" or response=="/index.html" or response=="/main_en.html" or
response=="/en"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: text/html; charset=utf-
8\r\n".encode())
        connectionSocket.send("\r\n".encode())
connectionSocket.send(hfile.encode())

elif(response=="/img/gf.jpg"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: image/jpg;\r\n".encode())
connectionSocket.send("\r\n".encode())         connectionSocket.send(hfile6)
     elif (response ==
"/img/hf.jpg"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: image/jpg;\r\n".encode())
connectionSocket.send("\r\n".encode())         connectionSocket.send(hfile7)
     elif (response ==
"/img/af.jpg"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: image/jpg;\r\n".encode())
connectionSocket.send("\r\n".encode())         connectionSocket.send(hfile8)
     elif (response ==
"/bb.jpg"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: image/jpg;\r\n".encode())
connectionSocket.send("\r\n".encode())         connectionSocket.send(hfile4)
     elif (response ==
"/b.png"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: image/png;\r\n".encode())
connectionSocket.send("\r\n".encode())         connectionSocket.send(hfile3)

    # for sending arabic page
elif(response=="/ar"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: text/html; charset=utf-
```

```
8\r\n".encode())
        connectionSocket.send("\r\n".encode())
connectionSocket.send(hfile0.encode())



    # for sending html file and display it as request
elif(response=="/.html%20file"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: text/html; charset=utf-
8\r\n".encode())
        connectionSocket.send("\r\n".encode())
connectionSocket.send(hfile1.encode())
     elif (response ==
"/html_en.css"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: text/css;\r\n".encode())
connectionSocket.send("\r\n".encode())
connectionSocket.send(hfile2.encode())
     elif (response ==
"/html_ar.css"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: text/css;\r\n".encode())
connectionSocket.send("\r\n".encode())
connectionSocket.send(hfile9.encode())

    # for sending css code
elif(response=="/.css"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: text/css;\r\n".encode())
connectionSocket.send("\r\n".encode())
connectionSocket.send(hfile2.encode())

    # for sending image with png type
elif(response=="/.png"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: image/png;\r\n".encode())
connectionSocket.send("\r\n".encode())         connectionSocket.send(hfile3)

    # for sending image with jpg type
elif (response=="/.jpg"):
        connectionSocket.send("HTTP/1.1 200 OK\r\n".encode())
connectionSocket.send("Content-Type: image/jpg;\r\n".encode())
connectionSocket.send("\r\n".encode())         connectionSocket.send(hfile4)



        # opening cornell website
    elif  response == "/cr":
        connectionSocket.send("HTTP/1.1 307 Temporary Redirect
\r\n".encode())
        connectionSocket.send('Content-Type: text/html; charset=utf-
8\r\n'.encode())
connectionSocket.send("Location:
```

```
https://www.cornell.edu\r\n".encode())
connectionSocket.send('\r\n'.encode())
        print("cornell.edu website successfully connected\r\n")
break

        # opening stackoverflow website
elif response == "/so":
            connectionSocket.send("HTTP/1.1 307 Temporary Redirect
\r\n".encode())
            connectionSocket.send('Content-Type: text/html; charset=utf-
8\r\n'.encode())
connectionSocket.send("Location:http://www.stackoverflow.com\r\n".encode())
connectionSocket.send('\r\n'.encode())
            print("stackoverflow.com website successfully connected\r\n")
break

        # opening ritaj website
elif response == "/rt":
            connectionSocket.send('HTTP/1.1 307 Temporary Redirect
\r\n'.encode())
            connectionSocket.send('Content-Type: text/html; charset=utf-
8\r\n'.encode())
connectionSocket.send("Location:https://ritaj.birzeit.edu/register/\r\n".enco
de())
            connectionSocket.send('\r\n'.encode())
            print("ritaj.com website successfully connected\r\n")
break

else:
            print("hhh "+response+"hhh")
                connectionSocket.send("HTTP/1.1 404 Not Found\r\n".encode())
          connectionSocket.send("Content-Type: text/html; charset=utf-
8\r\n".encode())
            connectionSocket.send("\r\n".encode())
connectionSocket.send(hfile5.encode())
```

Html code

```
<!DOCTYPE
 html>
<html lang="en">
    <head>

        <meta charset="UTF-8">

        <!--for link logo tab -->
        <link  rel="icon" href="b.png">

        <!--put our tittle -->
        <title>ENCS3320-My Tiny Webserver 23/24</title>
        <!--for link our css code  -->
        <link rel="stylesheet" href="html_en.css">
     </head>
    <body>

    <!-- put welcome message with background and logo with diffrent colors ->
    <h1  class="title"><img style="padding-right: 16px;" width="32" height="32"
src="b.png">Welcome to our course <strong style="color: blue;padding-left:
4px"> Computer Networks, This is a tiny webserver </strong></h1>

    <!-- defining boxes to put our information and detalis  -->
    <div class="box">
        <img class="imgg" src="img/gf.jpg">
        <div class="student-info">
        <h2> Ghassan 1212397</h2>
        <p> Hay, I am study computer engineering in Birzeit University, i'm
done 76 academic hours in third year , to be honest in last three weeks im
dont see my friend as usually from projects,homeworks and exams ,it is make me
busy and cant play my video games, i wish to be it worth.    </p>
    </div>
</div>
    <div class="box">
        <img class="imgg" src="img/hf.jpg">
        <div class="student-info">
        <h2>hassan 1211705</h2>
            <p> I am currently studying computer engineering in my third year.
Throughout my academic journey, I have successfully completed many projects,
and assignments in my academic subjects. When I have some free time, I love
indulging in my favorite pastimes of watching and playing football games and
going to the gym.</p>
    </div>
</div>
  <div class="box" >
        <img class="imgg" src="img/af.jpg">
        <div class="student-info">
        <h2>Aws Shaheen 1212585</h2>
        <p>This is my third year majoring in computer engineering at birzeit
university.I am intrested in web development and I am working on bettering my
skills in it, this is one of many projects I worked on for university for
many subjects. In my free time I usually go to the gym and spend time with my
freinds.</p>     </div>
    </div>
```

```
 <!-- defining box to put summary of point 0-->
<div class="box">
        <div class="student-info">
         <h2>point 0 summary</h2>
        <p> in point 0 we open the link and conclude the content type is
doc/html and we need it to make the client or server what tybe of content we
send </p>
        </div>
</div>

<!--links for w3schools and local html file  -->
<a href="https://www.w3schools.com/python/python_strings.asp"
target="_blank">W3School</a>
    <p></p>
    <a href="C:\Users\gassa\OneDrive\Desktop\MY
UNIVERSTY\Python\webserver\index.html" target="_blank">local html file</a>
    </body>
</html>
```

Css code

```
            /* define to be our texts nice */
            .student-info {
display: flex;                  justify-
content: flex-start;                  align-
items: flex-start;                  /* gap:
16px; */                 flex-direction:
column;

            }
            /*defining for tittle with background  */
            .title {
                background: burlywood;
width: 1500px;                  height:
100px;                  display: flex;
justify-content: center;
align-items: center;
margin-bottom: 32px;
font-family: verdana;
font-size: 20px;
            }
            /* define to be our texts nice */
h2{
                    font-style: oblique;
font-weight: bolder;
                    margin-top: 5px; /* Adjust the margin as needed */
padding: 0;
                    margin-bottom: 0; /* Adjust the margin as needed */

            }

             /* Define styles for the boxes */
             .box {
                display: flex;

            border: 1px solid #fffcfc;
padding: 20px;              margin:
10px;             box-sizing: border-
box;             position: relative;
overflow: hidden;             gap:
16px;             border-radius: 16px;
z-index: 1;
             }
            /* defining to be the text clear  */
            .box::before {
content: "";
position: absolute;
top: 0;             left: 0;
width: 100%;
height: 100%;
            background-image: url('bb.jpg');
            background-repeat: no-repeat;
background-size: cover;
filter:blur(200px);             z-index: -
1;
```

```
        }

            /*defining for images sizes */
             .imgg              {
width: 150px;
border-radius: 16px;

             }
            /*defining for be elemants horizontal */
             .horizontal-container {
display: flex;             flex-direction:
row;
            }

             /*defining for background of body */
body {
             background-image:
url("bb.jpg");             background-
repeat: no-repeat;             background-
size: cover;
        }
             /* define to be our texts nice */
h1{
            text-align: center;
            font-family: 'Times New Roman', Times, serif;
        }
        /* define to be our texts nice */
p {
    font-family: 'Times New Roman', Times, serif;     font-
size: 25px;
    margin-top: 0; /* Adjust the margin as needed */
margin-bottom: 5px; /* Adjust the margin as needed */
color:#fffcfc;     font-style:inherit;     font-
weight:90;
}
```